

TITLE OF THE INVENTION  
DOCUMENT PROCESSING METHOD AND APPARATUS

## FIELD OF THE INVENTION

5       The present invention relates to a document processing technique which provides an editing function of, e.g., combining separately created document files into one document file.

## 10      BACKGROUND OF THE INVENTION

Different types of data such as characters, tables, and images require different structures which define the data and different editing operations for the data. Various application programs are provided in accordance with the type of data. The user uses different application programs for different types of data to be edited: for example, a character processing program in order to edit characters, a spreadsheet program in order to edit tables, and an image editing program in order to edit images.

In this manner, the user generally uses different application programs for different types of data to be edited. However, a document to be created by the user generally contains a plurality of types of data such as characters and tables or characters and images, rather than a document formed by only one type of data such as characters, tables, or images. To create a target

document containing a plurality of types of data, the user must print data by various applications using their printing functions, and combine printed materials in a desired order.

5 There is known a print preview function of, before actually printing on a paper sheet in printing data such as a document or image edited by an application program for editing a document or image, displaying an image to be printed in a printing layout  
10 on the display and presenting the preview to the user.

Application software which combines data generated by various applications into one document has been devised. The present applicant has filed a mechanism disclosed in Japanese Patent Application  
15 No. 2001-280756. This system has a function of displaying a hierarchical structure of chapters, pages, and the like as a document structure in a tree format simultaneously when displaying a preview of a loaded document. The user can newly add a chapter or page to  
20 an existing document by directly inserting files created by various applications to the tree display region by drag-and-drop.

In this system, however, the user cannot perform detailed settings such as a chapter structure when  
25 inserting an object such as a document file or image file to the tree display region in the preview window by drag-and-drop. For example, whether to insert a

dragged/dropped document as a chapter or page is automatically determined by the application (fixed such that a document is inserted as a chapter or an image file is inserted as a page), and cannot be designated  
5 by the user.

#### SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above situation, and has as its  
10 object to provide a document processing technique capable of easily setting the document structure of a document file when inserting the document file into a predetermined document file.

One aspect of the present invention implements a  
15 document processing method and apparatus for providing an editing function of, e.g., combining separately created document files into one document file. When an instruction to insert the second document file to the first document file displayed in the display region is  
20 input, candidates for the insertion method of the second document file in inserting the second document file to the first document file are presented in accordance with the input instruction. The second document file is inserted to the first document file on  
25 the basis of the insertion method selected from the presented candidates.

According to a preferred aspect of the present

invention, the document structure of the first document file is displayed as a tree, and an instruction for the insertion position of the second document file in the document structure is input.

5       Other features and advantages of the present invention will be apparent from the following descriptions taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures  
10      thereof.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification,  
15      illustrate embodiments of the invention and, together with the descriptions, serve to explain the principle of the invention.

Fig. 1 is a block diagram showing the software configuration of a document processing system according  
20      to an embodiment;

Fig. 2A is a block diagram showing an example of a hardware configuration which implements the document processing system according to the embodiment;

Fig. 2B is a view showing a structure example of  
25      a mouse according to the embodiment;

Figs. 3A and 3B are views showing an example of a book file structure;

Figs. 4A and 4B are tables showing a list of book attributes;

Fig. 5 is a table showing a list of chapter attributes;

5 Fig. 6 is a table showing a list of page attributes;

Fig. 7 is a flow chart showing an example of a sequence of opening a book file;

10 Fig. 8 is a flow chart showing an example of a sequence of importing an electronic document file to a book file;

Fig. 9 is a flow chart showing an example of a sequence of generating an electronic document file;

15 Fig. 10 is a view showing an example of a user interface window when an existing book file is opened;

Fig. 11 is a view showing an example of a user interface window when a new book file is opened;

20 Fig. 12 is a block diagram showing an example of the software configuration of a client-server document processing system;

Fig. 13 is a view for explaining a conventional file insertion method;

Fig. 14 is a view showing an example of a file insertion result according to the embodiment;

25 Fig. 15 is a view showing a display example of document structure type candidates according to the embodiment;

Fig. 16 is a view showing a menu display example for each drop position when a single file is dropped to the tree portion by drag-and-drop using a right click;

5 Figs. 17, 18A, 18B, 19A, 19B, 20A, and 20B are views showing examples of a result of inserting a single file by drag-and-drop using a right click;

10 Fig. 21 is a view showing a menu display example for each drop position when a plurality of files are dropped to a tree portion 1101 by drag-and-drop using a right click; and

Figs. 22A, 22B, 23A to 23C, 24A to 24C, and 25A to 25C are views showing examples of a result of inserting a plurality of files by drag-and-drop using a right click.

15

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention are described in detail in accordance with the accompanying drawings.

20 <System Outline>

The outline of a document processing system according to an embodiment of the present invention will be described with reference to Figs. 1 to 12.

25 This document processing system converts a data file created by a general application into an electronic document file by an electronic document writer. A bookbinding application provides a function of editing

the electronic document file. Details of the system will be explained below.

<System Configuration and Operation>

Fig. 1 is a block diagram showing the software configuration of the document processing system according to the embodiment of the present invention.

The document processing system is implemented by a digital computer 100 as an embodiment suited to an information processing apparatus according to the embodiment. A general application 101 is an application program which provides functions such as wordprocessing, spreadsheet, photo-retouch, draw, paint, presentation, and text editing. The general application 101 has a printing function. Such application programs utilize a predetermined interface provided by the OS (Operating System) in printing application data such as created document data or image data. To print created data, the application 101 transmits an output command (GDI function) which is determined in advance for the output module (not shown) of the OS providing the interface and has an OS-dependent format. The output module which has received the output command converts the command into a format processible by an output device such as a printer, and outputs the converted command (DDI function). Since the format processible by the output device changes depending on the type of device, the

manufacturer, and the model, a device driver is provided for each device. The OS converts a command by using the device driver to generate printing data, and combines printing data by JL (Job Language) to generate 5 a print job. When the OS is Microsoft Windows®, the output module is a GDI (Graphical Device Interface) module.

An electronic document writer 102 is an improvement of the above-mentioned device drivers, and 10 is a software module provided to implement the document processing system. The electronic document writer 102 does not target a specific output device, and generates an electronic document file 103 by converting an output command into a format processible by a bookbinding 15 application 104 or printer driver 106 (to be described later). The format (to be referred to as an "electronic document format" hereinafter) converted by the electronic document writer 102 is not particularly limited as far as each document page can be expressed 20 by a detailed format. Of substantial standard formats, for example, the PDF format by Adobe Systems and the SVG format can be adopted as electronic document formats.

When the application 101 utilizes the electronic 25 document writer 102, the electronic document writer 102 is designated as a device driver used for output, and then caused to execute printing. An electronic

document file created by the electronic document writer 102 does not have a complete electronic document file format. For this reason, the electronic document writer 102 is designated as a device driver by the 5 bookbinding application 104, and executes conversion of application data into an electronic document file under the management of the bookbinding application 104.

The bookbinding application 104 completes a new incomplete electronic document file generated by the 10 electronic document writer 102 as an electronic document file having the following format. In case of necessity to definitely discriminate these files, a file created by the electronic document writer 102 will be called an "electronic document file", and an 15 electronic document file given a structure by the bookbinding application will be called a "book file". If these files need not be particularly discriminated, a document file generated by an application, an electronic document file, and a book file are called 20 "document files (or document data)".

As described above, the electronic document writer 102 is designated as a device driver, and the general application 101 prints the data. Application data is converted into an electronic document format in 25 pages (to be referred to as "logical pages" or "document pages" hereinafter) defined by the application 101. The converted data is stored as the

electronic document file 103 in a storage medium such as a hard disk. The hard disk may be the local drive of the computer which implements the document processing system according to the embodiment, or when 5 the computer is connected to a network, may be a drive provided on the network.

The bookbinding application 104 provides the user with a function of loading and editing the "electronic document file" or "book file" 103. The bookbinding 10 application 104 provides no function of editing the contents of each page, but a function of editing a chapter or book structure (to be described later) made up of pages as a minimum unit. As will be described later, the bookbinding application 104 has a function 15 of designating, for the book file 103, printing settings (printing method (double-sided/single-sided/bookbinding printing), printing layout (N-up/poster printing), color processing mode (color/monochrome), finishing setting 20 (ON/OFF of stapling/saddle stitch/punching and the like), additional information setting (watermark, user name, date, background, and the like), chaptering, paper insertion (index sheet, tabbed sheet, front cover, and back cover), the number of copies, sorting 25 designation, and the like).

To print the book file 103 edited by the bookbinding application 104, the bookbinding

application 104 activates an electronic document despooler 105. The electronic document despooler 105 reads out a designated book file from the hard disk. To print each page in a format described in the book 5 file, the electronic document despooler 105 generates an output command complying with the output module of the OS described above, and outputs the command to the output module (not shown). At this time, the printer driver 106 of a printer 107 used as an output device is 10 designated as a device driver. The output module converts the output command received using the designated printer driver 106 of the printer 107 into a device command interpretable by the printer 107. The device command is transmitted to the printer 107, which 15 prints an image corresponding to the command.

Fig. 2A is a block diagram showing the hardware configuration of the digital computer 100 shown in Fig. 1.

In Fig. 2A, a CPU 201 executes a program such as 20 an OS, the general application 101, or the bookbinding application 104 which is stored in the program ROM of a ROM 203 or loaded from a hard disk 211 to a RAM 202, and realizes the software configuration in Fig. 1 or flow chart sequence (to be described later). The RAM 25 202 functions as the main memory or work area of the CPU 201. A keyboard controller (KBC) 205 controls a key input from a keyboard 209 or an input from a mouse

209a serving as a pointing device. A CRT controller (CRTC) 206 controls the display on a display 210. The display 210 is not limited to a CRT, and may be a liquid crystal display or plasma display. A disk 5 controller (DKC) 207 controls access to the hard disk (HD) 211 or flexible disk (FD) which stores a boot program, various applications, font data, user files, editing files (to be described later), and the like. A PRTC 208 controls signal exchange with the connected 10 printer 107. An NC 212 is connected to a network, and executes communication control processing with another device connected to the network.

Fig. 2B is a view showing a structure example of the mouse 209a.

15 The mouse 209a in the embodiment is, e.g., a 2-button mouse. In Fig. 2B, reference numeral 209R denotes a right button; and 209L, a left button. In this specification, an operation of pressing either button once and releasing it within a short time will 20 be called "click". The click of the right button 209R will be called "right click", and that of the left button 209L will be called "left click".

<Electronic Document Data Format>

Before a detailed description of the bookbinding 25 application 104, the "book file" data format will be explained.

The book file has a three-layered structure

similar to a paper-medium book. The upper layer is called a "book", resembles one book, and defines the attributes of the entire book. The intermediate layer corresponds to a chapter in the book, and is also  
5 called a "chapter". As for each "chapter", its attributes can be defined. The lower layer is a "page", and corresponds to each page defined by an application program. As for each "page", its attributes can be defined. One "book" may contain a  
10 plurality of "chapters", and one "chapter" may contain a plurality of "pages".

Fig. 3A is a block diagram schematically showing an example of the book file format. In the book file of this example, a "book", "chapter", and "page" are  
15 represented by corresponding nodes. One book file contains one "book". The "book" and "chapter" are a concept for defining a book structure, and contain, as entities, defined attribute values and links to lower layers. The "page" has, as an entity, data of each  
20 "page" output from an application program. In addition to an attribute value, the "page" contains the entity of a document page (document page data) and a link to each document page data. In some cases, a printing page to be output onto a paper medium or the like  
25 contains a plurality of document pages. This structure is displayed not by a link but by an attribute in the "book", "chapter", or "page" layer.

In Fig. 3A, a book 301 defines a book attribute, and is linked to two chapters 302A and 302B. These links display that the chapters 302A and 302B are contained in the book 301. The chapter 302A is linked 5 to pages 303A and 303B, which represents that the chapter 302A contains these pages. The pages 303A and 303B define page attribute values, and contain links to document page data (1) and (2) serving as entities. These links represent document data (1) and (2) of 10 document page data 304 shown in Fig. 3B, and display that the entities of the pages 303A and 303B are document page data (1) and (2).

Similarly, the chapter 302B contains pages 303C and 303D, and the pages 303C and 303D contain links to 15 document page data (3) and (4) serving as entities. These links represent document data (3) and (4) of document page data 304 shown in Fig. 3B, and display that the entities of the pages 303C and 303D are document page data (3) and (4).

20 Figs. 4A and 4B are tables for explaining the book attribute 301 according to the embodiment.

In Figs. 4A and 4B, as for an item which can be defined repetitively in a lower layer, the attribute value of the lower layer is preferentially adopted. As 25 for an item contained in only the book attribute 301, a value defined in the book attribute 301 is effective over the book. An item repetitively defined in a lower

layer is a default value used when this item is not defined in the lower layer. Each item shown in Figs. 4A and 4B do not correspond to one concrete item, but may contain a plurality of relevant items.

5 Fig. 5 is a table for explaining the chapter attribute according to the embodiment, and Fig. 6 is a table for explaining the page attribute according to the embodiment. The relationship between the chapter attribute and the page attribute is the same as that  
10 between the book attribute and the lower layer attribute.

In Figs. 4A and 4B, items unique to the book attribute 301 are six items: "printing method", "details of bookbinding", "front/back cover", "index sheet", "slip sheet", and "chaptering". These items are defined over the book.

As "printing method" attributes, three values: "single-sided printing", "double-sided printing", and "bookbinding printing" can be designated. "Bookbinding printing" is a method of printing data in a format which allows bookbinding by bundling a separately designated number of paper sheets, folding the bundle into two, and binding the bundle. As "detailed bookbinding" attributes, the "opening direction" and  
20 the "number of paper sheets to be bundled" can be designated when the "printing method" designates "bookbinding printing".

The "front/back cover" attribute includes designation of adding paper sheets serving as front and back covers when an electronic document file combined as a book is printed, and designation of contents to be

5 printed on the added paper sheets. The "index sheet" attribute includes designation of inserting a tabbed index sheet separately prepared in a printing apparatus for chaptering, and designation of contents to be printed on the index (tabbed) portion. This attribute

10 becomes effective when a printing apparatus for use is equipped with an inserter having an inserting function of inserting a paper sheet prepared separately from a printing paper sheet into a desired position, or when a plurality of sheet cassettes can be used. This also

15 applies to the "slip sheet" attribute. The "slip sheet" attribute includes designation of inserting a paper sheet fed from an inserter or sheet feed cassette for chaptering, and designation of a sheet feed source when a slip sheet is inserted.

20 The "chaptering" attribute includes designation of whether to use a new paper sheet, use a new printing page, or do nothing particular at a chapter break. In "single-sided printing", the use of a new paper sheet and the use of a new printing page are the same. In

25 "double-sided printing", a continuous chapter is not printed on one paper sheet if "the use of a new paper sheet" is designated, but may be printed on the obverse

and reverse of one paper sheet if "the use of a new printing page" is designated.

As for the chapter attribute shown in Fig. 5, there is no item unique to the chapter, and all items 5 overlap those of the book attribute. If the definition of the chapter attribute is different from that of the book attribute, a value defined by the chapter attribute precedes. Items common to only the book and chapter attributes are five items: "paper size", "paper 10 direction", "N-up printing designation", "enlargement/reduction", and "discharge method". The "N-up printing designation" attribute is an item for designating the number of document pages contained in one printing page. Layouts which can be designated are 15 1 x 1, 1 x 2, 2 x 2, 3 x 3, 4 x 4, and the like. The "discharge method" attribute is an item for designating whether to staple discharged paper sheets. The effectiveness of this attribute depends on whether the printing apparatus has a stapling function.

20 In the page attribute shown in Fig. 6, items unique to the page attribute are "page rotation designation", "zoom", "layout designation", "annotation", and "page division". The "page rotation designation" is an item for designating the rotation angle when a document page is laid out on a printing 25 page. The "zoom" attribute is an item for designating the zoom ratio of a document page. The zoom ratio is

designated based on a virtual logical page region size = 100%. The virtual logical page region is a region occupied by one document page when document pages are laid out in accordance with N-up designation or the 5 like. For example, the virtual logical page region is a region corresponding to one printing page for 1 x 1, and a region obtained by reducing each side of one printing page to about 70% for 1 x 2.

Attributes common to the above-described "book", 10 "chapter", and "page" are a "watermark" attribute and "header/footer" attribute. The "watermark" is a separately designated image or character string printed over data created by an application. The "header and footer" are watermarks printed at the upper and lower 15 margins of each page. For the "header and footer", items such as a page number, and time and date which can be designated by variables are prepared. Contents which can be designated by the "watermark" attribute and "header/footer" attribute are common to the 20 "chapter" and "page", but are different in the "book". The "book" can set the contents of the "watermark" and "header/footer", and designate how to print a "watermark" or "header/footer" throughout the book. To the contrary, the "chapter" and "page" can designate 25 whether to print a "watermark" or "header/footer" set by the book on the chapter or page.

<Book File Generation Sequence>

The "book file" serving as an electronic document file which is given a structure by the bookbinding application 104 has the above-described structure and contents. A sequence of creating a book file by the 5 bookbinding application 104 and electronic document writer 102 will be explained. Creation of a book file is realized as part of book file editing operation by the bookbinding application 104.

Fig. 7 is a flow chart for explaining a sequence 10 of opening a book file by the bookbinding application 104 according to the embodiment.

In step S701, the bookbinding application 104 determines whether a book file to be opened is one to be newly created or an existing one. If YES in step 15 S701, the processing advances to step S702, and the bookbinding application 104 newly creates a book file containing no chapter. In the example shown in Fig. 3, the newly created book file is a book node which has only the book node 301 without any link to a chapter 20 node. As the book attribute, a set of attributes prepared in advance for creation of a new book file are applied. The processing advances to step S704, and the bookbinding application 104 displays a UI (User Interface) window for editing the new book file.

25 Fig. 11 shows an example of the UI window when a book file is newly created. In this case, a UI window 1100 does not display any information because the book

file does not have any substantial content.

If NO in step S701, the processing advances to step S703, and the bookbinding application 104 opens a designated book file and displays a UI (User Interface) 5 window in accordance with the structure, attribute, and contents of the book file.

Fig. 10 is a view showing an example of the displayed UI window.

The UI window 1100 has a tree portion 1101 10 representing a book structure, and a preview portion 1102 displaying a state to be printed. The tree portion 1101 displays (displays a tree) chapters contained in the book and pages contained in each chapter by a tree structure as shown in Fig. 3A. The 15 tree portion 1101 displays page numbers, and the page numbers represent document page numbers. The preview portion 1102 displays reduced printing page contents. The display order reflects the book structure.

Application data converted into an electronic 20 document file by the electronic document writer 102 can be added as a new chapter or page to the open book file (regardless of a new book file or existing book file). This function is called an "electronic document import 25 function". An electronic document is imported to the book file newly created by the sequence shown in the flow chart of Fig. 7, thereby giving an entity to the book file. This function is activated by drag-and-drop

operation of application data to the window of Fig. 10. Note that drag-and-drop operation is an operation of positioning a mouse pointer to an object (i.e., application data), moving the mouse while pressing the 5 left button 209L of the mouse, and releasing the mouse button at a target position. The mouse in Fig. 2B is a right-handed mouse. For a left-handed mouse, drag-and-drop operation is an operation of moving (dragging) the mouse while pressing the right button 10 209R of the mouse and selecting an object, and releasing (dropping) the mouse button at a target position. By drag-and-drop operation, new application data can be added to a book file which has already been opened. In the bookbinding application, drag-and-drop 15 operation adds a new chapter when the object is a document file, and a new page when the object is an image file.

Fig. 8 is a flow chart showing an electronic document import sequence according to the embodiment.

20 An application program which has generated designated application data is activated. The electronic document writer 102 is designated as a device driver, and prints out application data. The electronic document writer 102 generates electronic 25 document data from the drawing function of application data input via the OS (step S801). After conversion into the electronic document data, the processing

advances to step S802, and the bookbinding application 104 determines whether the converted data is image data. This determination can be achieved based on the file extension of application data under the Windows 5 OS. For example, an extension "bmp" represents Windows bitmap data; "jpg", JPEG-compressed image data; and "tiff", TIFF-format image data. If YES in step S802, processing in step S801 can be skipped because an electronic document file can be directly generated from 10 image data without activating an application in step S801.

If NO in step S802, the processing advances to step S803, and the bookbinding application 104 adds the electronic document file generated in step S801 as a 15 new chapter to the book of a currently open book file. In this case, as for the chapter attribute, a book attribute value is copied for an attribute common to a book attribute, and a different attribute is set to a default value prepared in advance.

20 If YES in step S802, the processing advances to step S804, and the bookbinding application 104 adds each document page contained in the electronic document file generated in step S801 to a designated chapter without adding no new chapter in principle. For a file 25 in which a book file is newly created, a new chapter is created, and each page of the electronic document file is added as a page belonging to the chapter. As for

the page attribute, an attribute common to an upper layer attribute is given the attribute value of the upper layer attribute, and an attribute which is defined in application data and inherited to the 5 electronic document file is given a value defined in the application data. For example, when "N-up printing designation" is defined in application data, the page inherits this attribute value. In this way, a new book file is created, or a new chapter is added.

10 Fig. 9 is a flow chart showing a sequence of generating an electronic document file by the electronic document writer 102 in step S801 of Fig. 8.

In step S901, a new electronic document file is created and opened. The processing advances to step 15 S902, and an application corresponding to designated application data is activated. The electronic document writer 102 is set as a device driver to transmit an output command to the output module of the OS. The output module converts the received output command into 20 data of the electronic document format by the electronic document writer, and outputs the converted data. The output destination is the electronic document file opened in step S901. The processing advances to step S903 to determine whether all 25 designated data have been converted. If YES in step S903, the processing advances to step S904 to close the electronic document file.

The electronic document file generated by the electronic document writer 102 is a file containing document page data entities shown in Fig. 3B.

<Editing of Book File>

5 As described above, a book file can be created from application data. The generated book file allows editing a chapter and page as follows.

(1) New document

(2) Delete

10 (3) Copy

(4) Cut

(5) Paste

(6) Move

(7) Change chapter name

15 (8) Reassign page number/name

(9) Insert cover

(10) Insert slip sheet

(11) Insert index sheet

(12) Page layout of each document page

20 In addition, an operation of canceling executed editing operation, and an operation of restoring canceled operation can be performed. These editing functions enable editing operations such as consolidation of a plurality of book files,

25 rearrangement of chapters and pages within a book file, delete of chapters and pages within a book file, layout change of a document page, and insertion of a slip

sheet and index sheet. By these operations, operation results are reflected in attributes shown in Figs. 4A to 6 or on a book file structure. For example, a blank page is inserted to a designated portion by an  
5 operation of newly adding a blank page. The blank page is processed as a document page. If the layout of a document page is changed, the change contents are reflected in attributes such as the printing method, N-up printing, front/back cover, index sheet, slip  
10 sheet, and chaptering.

<Output of Book File>

The ultimate goal of a book file created and edited in the above manner is to print it out. The user selects a file menu from the UI window 1100 of the  
15 bookbinding application shown in Fig. 10, and selects printing from this menu. The book file is then printed out from a designated output device. At this time, the bookbinding application 104 creates a job ticket from a currently open book file, and transfers the job ticket  
20 to the electronic document despooler 105. The electronic document despooler 105 converts the job ticket into an OS output command, e.g., a Windows GDI command, and transmits the command to an output module, e.g., GDI. The output module generates a command  
25 complying with an output device by a designated printer driver 106, and transmits the command to the output device.

The job ticket is data with a structure whose minimum unit is a document page. The data structure of the job ticket defines the layout of a document page on paper. One job ticket is issued for one job.

5     A document node is set at the top of the structure, and defines the attribute of the whole document such as double-sided printing/single-sided printing. A paper node belongs to the document node, and contains attributes such as the identifier of paper for use and

10    designation of a feed port in the printer. A sheet node to be printed on the paper belongs to each paper node. One sheet corresponds to one printing paper sheet. A printing page (physical page) belongs to each sheet. One physical page belongs to one sheet for

15    single-sided printing, and two physical pages belong to one sheet for double-sided printing. A document page to be laid out on a physical page belongs to the physical page. The physical page attribute contains a document page layout.

20       The electronic document despooler 105 converts the job ticket into an output command to the output module.

<Another System Configuration>

The outline of the document processing system  
25 according to the embodiment has been described. This system is of a stand-alone type. A server-client system as an extension of the stand-alone system also

creates and edits a book file by almost the same arrangement and sequence. A book file and printing processing are managed by the server.

Fig. 12 is a block diagram showing the  
5 configuration of a server-client document processing system according to the embodiment.

A client document processing system 1200 is constituted by adding to the stand-alone system a DOMS (Document Output Management Service) driver 109 serving  
10 as a client module, a DOMS print service module 110, and a DS (Document Service) client module 108. The client document processing system 1200 is connected to a document management server 1201, centralized printing management server 1202, and print server 1203. These  
15 servers are generally connected to the client document processing system 1200 via a network. When the servers also function as clients, they are connected by interprocess communication which simulates communication between networks.

20 The document management server 1201 and centralized printing management server 1202 are connected to the client in Fig. 12, but only either one may exist on the network. If the connected server is the document management server, a document management  
25 server-client system 1201SC including a client module is added to the stand-alone document management system. If the connected server is the centralized printing

management server 1202, a printing management server-client system 1202SC including a client module is added.

The document management server 1201 stores a book 5 file created and edited by the bookbinding application 104. To manage a book file by the document management server 1201, the book file is saved in a database 1211 of the document management server 1201 instead of or in addition to the local HD of a client PC. Save and read 10 of a book file between the bookbinding application 104 and the document management server 1201 are done via the DS client module 108 and a DS core 1212.

The centralized printing management server 1202 manages printing of a book file stored in the client 15 document processing system 1200 or document management server 1201. A printing request from the client is transmitted to a DOMS WG server module 1221 of the centralized printing management server 1202 via the DOMS driver 109 and DOMS print service module 110. To 20 print a book file by the printer 107 of the client, the centralized printing management server 1202 transfers electronic document data to the electronic document despooler 105 via the DOMS print service module 110 of the client.

25 To print a book file by the print server 1203, the centralized printing management server 1202 transmits electronic document data to a DOMS print

service module 1231 of the print server 1203. For example, the centralized printing management server 1202 executes security check on the qualification of a user who has issued a printing request for a saved book 5 file, or saves the printing processing log. In this fashion, the document processing system can be implemented as both a stand-alone system and client-server system.

<Preview Display Contents>

10 As described above, when a book file is opened by the bookbinding application, the user interface window 1100 shown in Fig. 10 is displayed. The tree portion 1101 displays a tree representing the structure of the open book (to be referred to as a "book of interest" 15 hereinafter). At the printing preview portion 1102, three display methods are prepared in accordance with designation by the user.

The first method is a "document view mode" in which document pages are directly displayed. In the 20 "document view mode", the contents of document pages belonging to the book of interest are reduced and displayed. The display of the preview portion 1102 does not reflect any document layout.

The second method is a "printing view mode". In 25 the "printing view mode", the preview portion 1102 displays document pages in a format which reflects the layout of them.

The third method is a "simple printing view mode". In the "simple printing view mode", the contents of document pages are not reflected in the display of the preview portion 1102, but only the 5 layout is reflected.

<Insertion of File by Drag-and-drop>

A file insertion method by drag-and-drop according to the embodiment will be explained.

When a data file created by the general 10 application is inserted by drag-and-drop to the tree portion 1101 of the user interface window 1100, an electronic document file 103 is created by the electronic document writer 102 and added to an existing book file.

15 At this time, in the above-described system, target files (DTPtest3.doc and DTPtest4.doc) are selected by a left click of the right-handed mouse, and inserted by designating a position desired by the user at the tree portion 1101 by drag-and-drop, as shown in 20 the example of Fig. 13.

Whether to insert one or a plurality of files as chapters or pages is automatically determined by the bookbinding application 104 on the basis of whether the target file is general application data or an image 25 file. To reset the chapter structure to one desired by the user after inserting data to the tree portion 1101, the user must perform corresponding operation. This

operation is cumbersome especially when data to be inserted is a file formed by a plurality of pages or a plurality of files are inserted at once.

To prevent this, an object is inserted by the  
5 following operation in the embodiment. Note that the following mouse operation is based on a right-handed mouse.

(Insertion of File by Left Click)

Fig. 14 is a view showing an example of inserting  
10 a file to the existing electronic document file 103 by drag-and-drop using a left click.

In this case, when files (DTPtest3.doc and DTPtest4.doc) are dropped to a branch 1301 representing the chapter name in Fig. 13 ("title 1" in the example  
15 of Fig. 13), these files are inserted as preceding chapters before the chapter "title 1", as represented by 1401 in Fig. 14. When files (scanner.jpg and scanner3.jpg) are dropped to a page in Fig. 13 ("2-1" page in the example of Fig. 13), these files are  
20 inserted as pages subsequent to the page "2-1", as represented by 1402 in Fig. 14. File insertion by drag-and-drop using a left click is performed in accordance with document structure information of a drop position (i.e., an insertion position) with  
25 default settings determined in advance by the bookbinding application 104, as described above. In the embodiment, the document structure of a document

file is formed by at least a chapter in an upper layer and a page in the lower layer.

(Insertion of File by Right Click)

When a plurality of files are dragged and dropped

5 by using a right click, the bookbinding application 104 determines the type of drag-and-drop (right or left click). If the bookbinding application 104 determines drag-and-drop using a right click, the application 104 displays, in a menu dialog, document structure type

10 candidates for a file to be inserted, as shown in Fig. 15. For example, the document structure type includes "create chapters by the number of files" (1501), "insert all files to one chapter" (1502), and "insert all files as pages" (1503). Any one type can

15 be selected from these candidates by using the mouse or keyboard. The bookbinding application 104 displays an item selected by the user in an inverted color. The example of Fig. 15 illustrates a dialog displayed when drag-and-drop operation using a right click is

20 performed while a plurality of files are selected.

When drag-and-drop operation using a right click is performed while a single file is selected, a menu dialog for selecting either "insert all files to one chapter" or "insert all files as pages" is displayed.

25 That is, if drag-and-drop operation is done, the bookbinding application 104 determines whether this operation is drag-and-drop using a right click or

drag-and-drop using a left click, and if determining that the operation is drag-and-drop using a right click, determines whether the operation is drag-and-drop of a plurality of files or drag-and-drop of a single file. If the bookbinding application 104 determines that the operation is drag-and-drop by a right click that selects a plurality of files, the application 104 displays the menu dialog shown in Fig. 15. If the bookbinding application 104 determines that the operation is drag-and-drop by a right click that selects a single file, the application 104 displays a menu dialog for selecting either "insert all files to one chapter" or "insert all files as pages". The display of the menu dialog changes in accordance with the drop position, which will be described later.

In inserting a plurality of files, if "create chapters by the number of files" (1501) is selected as the first document structure type, the bookbinding application 104 newly generates, from the insertion position, chapters corresponding to the number of files to be inserted, and the pages of each file are arranged in a corresponding generated chapter. If "insert all files to one chapter" (1502) is selected as the second document structure type, the bookbinding application 104 newly generates one chapter at the insertion position, and the pages of files to be inserted are arranged in the chapter. If "insert all files as

pages" (1503) is selected as the third document structure type, the bookbinding application 104 directly arranges pages at the insertion position.

(Insertion of Single File by Right Click)

5 Fig. 16 is a view showing a menu display example at each drop position when a single file is dropped to the tree portion 1101 by drag-and-drop using a right click.

As shown in Fig. 16, when a single file is  
10 dropped to the tree portion 1101 by a right click, a menu corresponding to the drop position is displayed on the basis of the menu shown in Fig. 15.

More specifically, the bookbinding application  
104 controls to change selectable items in accordance  
15 with the drop position and display the menu dialog. In the embodiment, all menus which cannot be selected depending on the drop position are grayed out. In this case, the number of files to be inserted is one, and "create chapters by the number of files" 1501 is grayed  
20 out so as not to select it. When a file is to be inserted before the first chapter, as represented by 1601 in Fig. 16, "insert all files as pages" 1503 is grayed out so as not to select it.

In short, the embodiment presents document  
25 structure type candidates corresponding to at least either the file insertion position or the number of selected document files.

An insertion method when a single file is dropped to each of positions 1601, 1602, 1603, and 1604 will be explained.

A case wherein one file is inserted to the 5 position 1601 by a right click will be described. In this case, a single file is dragged and dropped immediately before the first chapter ("title 1"), and only "insert all files to one chapter" (1502) can be selected from the displayed menu. If the item 1502 is 10 selected, a chapter is newly created at the top of the tree portion, and the pages of a new file are inserted to the newly created chapter, as represented by 1701 in Fig. 17.

When a file is inserted to the position 1602, 15 1603, or 1604, the two items "insert all files to one chapter" (1502) and "insert all files as pages" (1503) can be selected from the menu.

If "insert all files to one chapter" (1502) is selected from the menu upon dropping a file to the 20 position 1602 between a chapter name ("sample shaping 1") and the first page (2-1), the dropped file (DTP test 3) is inserted as a new chapter by the bookbinding application 104 immediately before the first page (2-1) of the dropping destination chapter (sample shaping 1), 25 as represented by 1801 in Fig. 18A. The document page (2-1) in the dropping destination chapter is moved as a "new chapter" after the inserted chapter. That is, the

existing chapter (in this case, "sample shaping 1") is separated. If a file is dropped to the position 1602 and "insert all files as pages" (1503) is selected from the menu, the file is inserted as pages by the  
5 bookbinding application 104 from the start of the existing chapter to which the file has been dropped, as represented by 1802 in Fig. 18B.

A case wherein a file is dropped to the position 1603 between a chapter name ("purpose of DTP test") and  
10 the final page (4-1) of the immediately preceding chapter ("title 2") will be explained. At this time, if "insert all files to one chapter" (1502) is selected from the displayed menu, a new chapter is created at the drop position by the bookbinding application 104,  
15 and pages are arranged in this chapter, as represented by 1901 in Fig. 19A. If a file is dropped to the position 1603 and "insert all files as pages" (1503) is selected from the menu, the file is directly inserted as pages to the drop position in the existing chapter  
20 by the bookbinding application 104, as represented by 1902 in Fig. 19B.

If a file is dropped to the position 1604 between pages (5-7 and 5-8) in the chapter name ("purpose of DTP test") and "insert all files to one chapter" (1502)  
25 is selected from the displayed menu, a new chapter is created at the drop position by the bookbinding application 104, pages are arranged in this chapter,

and pages after the drop position are so rearranged as to belong to another new chapter, as represented by 2001 in Fig. 20A. If a file is dropped to the position 1604 and "insert all files as pages" (1503) is selected 5 from the menu, the file is directly inserted as pages to the drop position in the existing chapter by the bookbinding application 104, as represented by 2002 in Fig. 20B.

(Insertion of Plurality of Files by Right Click)

10 Insertion of a plurality of files by using a right click will be described.

Fig. 21 is a view showing a menu display example at each drop position when a plurality of files are dropped to the tree portion 1101 by drag-and-drop using 15 a right click. Similar to a single file, when files are dragged and dropped to an insertion position by a right click, a menu dialog as shown in Fig. 15 is displayed by the bookbinding application 104, and the user can select an arbitrary item from the dialog.

20 When a plurality of files are to be inserted at the top of the tree portion, as represented by 2101, "insert all files as pages" (1503) is grayed out so as not to select it.

When a plurality of selected files include an 25 image file, the image file is arranged at the end of inserted objects.

Cases wherein a plurality of files are dropped to

positions 2101, 2102, 2103, and 2104 will be explained.

A case wherein a plurality of designated files ("DTPtest3.doc", "DTPtest4.doc", and "scanner.jpg") are inserted to the position 2101 which is the start 5 position of the document will be first described. In this case, "insert all files as pages" (1503) cannot be selected, as described above, and only either "create chapters by the number of files" (1501) or "insert all files to one chapter" (1502) can be selected. If the 10 item 1501 is selected from the menu, a chapter having a name corresponding to the file name of each file is created for the file by the bookbinding application 104, and pages corresponding to each chapter are inserted, as represented by 2201 in Fig. 22A. If the 15 item 1502 is selected from the menu, all files are inserted as pages to a chapter newly created by the bookbinding application 104, as represented by 2202 in Fig. 22B.

A case wherein a plurality of files are dropped 20 to the position 2102 between the chapter name ("sample shaping 1") and the first page (2-1) will be described. In this case, if "create chapters by the number of files" (1501) is selected from the menu, a chapter having a name corresponding to the file name of each 25 file is created for the file from a position immediately before the first page (2-1) of the chapter "sample shaping 1" by the bookbinding application 104,

and pages corresponding to each chapter are inserted, as represented by 2301 in Fig. 23A. In this case, a new chapter is created after the added chapters, and document pages in the chapter "sample shaping 1" are  
5 moved to the created chapter.

If a plurality of files are dropped to the position 2102 and "insert all files to one chapter" (1502) is selected from the menu, a new chapter is created at a position immediately before the first page  
10 (2-1) of the chapter "sample shaping 1" by the bookbinding application 104, and all files are added as pages to the chapter, as represented by 2302 in Fig. 23B. Another new chapter is created after the added chapter, and document pages at the insertion  
15 position are moved to the created chapter.

If a plurality of files are dropped to the position 2102 and "insert all files as pages" (1503) is selected from the menu, all files are inserted as pages from a position immediately before the first page of  
20 the chapter "sample shaping 1" by the bookbinding application 104, as represented by 2303 in Fig. 23C.

A case wherein a plurality of files are dropped to the position 2103 between the chapter name ("purpose of DTP test") and the final page (4-1) of the  
25 immediately preceding chapter ("title 2") will be described. In this case, if "create chapters by the number of files" (1501) is selected from the displayed

menu, a chapter having a name corresponding to the file name of each file is created for the file from a position immediately before the first page (5-1) of the chapter "purpose of DTP test" by the bookbinding application 104, and pages corresponding to each chapter are inserted, as represented by 2401 in Fig. 24A. If a plurality of files are dropped to the position 2103 and "insert all files to one chapter" (1502) is selected from the displayed menu, a new chapter is created at the drop position by the bookbinding application 104, and all files are added as pages to the chapter, as represented by 2402 in Fig. 24B. If a plurality of files are dropped to the position 2103 and "insert all files as pages" (1503) is selected from the displayed menu, all files are inserted as pages at the end of the dropping destination chapter by the bookbinding application 104, as represented by 2403 in Fig. 24C.

A case wherein a plurality of files are dropped to the position 2104 between pages (5-7 and 5-8) in the chapter name ("purpose of DTP test") will be described. In this case, if "create chapters by the number of files" (1501) is selected from the displayed menu, a chapter having a name corresponding to the file name of each file is created for the file at the drop position by the bookbinding application 104, and pages corresponding to each chapter are inserted, as

represented by 2501 in Fig. 25A. In this case, chapters corresponding to the number of inserted files are created and added from the drop position. The remaining document pages at the drop position are moved 5 to a "new chapter" subsequent to the inserted chapters.

If a plurality of files are dropped to the position 2104 and "insert all files to one chapter" (1502) is selected from the displayed menu, a new chapter is created at the drop position by the 10 bookbinding application 104, and all files are inserted as pages to the chapter, as represented by 2502 in Fig. 25B. At this time, pages after the drop position are so rearranged as to belong to another new chapter. If a plurality of files are dropped to the position 15 2104 and "insert all files as pages" (1503) is selected from the displayed menu, the files are inserted as pages to the drop position by the bookbinding application 104, as represented by 2503 in Fig. 25C.

The above file insertion processing is summarized 20 as follows.

The document structure of a book file serving as the first document file is displayed as a tree. The user is prompted to select as the second document file a file to be inserted to the book file displayed as a 25 tree. The user is prompted to designate the insertion position of the second document file in the displayed document structure of the first document file, thus

inputting the designation. In response to the designation of the insertion position, the bookbinding application 104 presents, to the user, document structure type candidates for the second document file 5 in inserting the second document file to the insertion position. If the user selects any document structure type from the presented candidates, the bookbinding application 104 inserts the second document file to the insertion position on the basis of the selected 10 document structure type.

According to the above-described embodiment, a menu which allows the user to select how to insert a file is displayed in inserting the file by drag-and-drop. The user can easily insert the file in 15 a desired format. At the same time as file insertion, the chapter structure can be arranged, reducing the labor of the user and increasing the creation efficiency.

In the above-described embodiment, a 2-button 20 mouse as shown in Fig. 2 is adopted, and file insertion operation by drag-and-drop is changed between left and right clicks. In general, assignment of a predetermined file operation to a button can be changed by settings. That is, assignment to each button in the 25 above-described embodiment is merely an example which exhibits the best conceivable mode at present. Hence, contents assigned to each button of the mouse in the

above-described embodiment can be exchanged between the right and left buttons.

Drag-and-drop is not limited to mouse operation, and the same processing as drag-and-drop operation 5 using a right click can also be executed by drag-and-drop using a left click while pressing the "Shift" key of the keyboard.

Assume that drag-and-drop using a right click is defined as the first operation processing, and 10 drag-and-drop using a left click is defined as the second operation processing. The document management application (bookbinding application 104) according to the present invention controls to automatically add a target file to an existing book file when the second 15 operation processing is done, and to display a menu for designating how to add a target file to an existing book when the first operation processing is done.

Further, various types of mouses are available, and the pointing device also includes a track ball in 20 addition to the mouse. The above-described file operation can also be assigned using such pointing device different from that in the embodiment.

#### (Other Embodiments)

Note that the present invention can be applied to 25 an apparatus comprising a single device or to system constituted by a plurality of devices.

Furthermore, the invention can be implemented by

supplying a software program, which implements the functions of the foregoing embodiments, directly or indirectly to a system or apparatus, reading the supplied program code with a computer of the system or 5 apparatus, and then executing the program code. In this case, so long as the system or apparatus has the functions of the program, the mode of implementation need not rely upon a program.

Accordingly, since the functions of the present 10 invention are implemented by computer, the program code installed in the computer also implements the present invention. In other words, the claims of the present invention also cover a computer program for the purpose of implementing the functions of the present invention.

15       In this case, so long as the system or apparatus has the functions of the program, the program may be executed in any form, such as an object code, a program executed by an interpreter, or script data supplied to an operating system.

20       Example of storage media that can be used for supplying the program are a floppy disk, a hard disk, an optical disk, a magneto-optical disk, a CD-ROM, a CD-R, a CD-RW, a magnetic tape, a non-volatile type memory card, a ROM, and a DVD (DVD-ROM and a DVD-R).

25       As for the method of supplying the program, a client computer can be connected to a website on the Internet using a browser of the client computer, and

the computer program of the present invention or an automatically-installable compressed file of the program can be downloaded to a recording medium such as a hard disk. Further, the program of the present

5 invention can be supplied by dividing the program code constituting the program into a plurality of files and downloading the files from different websites. In other words, a WWW (World Wide Web) server that

10 downloads, to multiple users, the program files that implement the functions of the present invention by computer is also covered by the claims of the present invention.

It is also possible to encrypt and store the program of the present invention on a storage medium

15 such as a CD-ROM, distribute the storage medium to users, allow users who meet certain requirements to download decryption key information from a website via the Internet, and allow these users to decrypt the encrypted program by using the key information, whereby

20 the program is installed in the user computer.

Besides the cases where the aforementioned functions according to the embodiments are implemented by executing the read program by computer, an operating system or the like running on the computer may perform

25 all or a part of the actual processing so that the functions of the foregoing embodiments can be implemented by this processing.

- Furthermore, after the program read from the storage medium is written to a function expansion board inserted into the computer or to a memory provided in a function expansion unit connected to the computer, a
- 5 CPU or the like mounted on the function expansion board or function expansion unit performs all or a part of the actual processing so that the functions of the foregoing embodiments can be implemented by this processing.
- 10 As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the
- 15 appended claims.